

Prior Art
Figure 1

00965249545

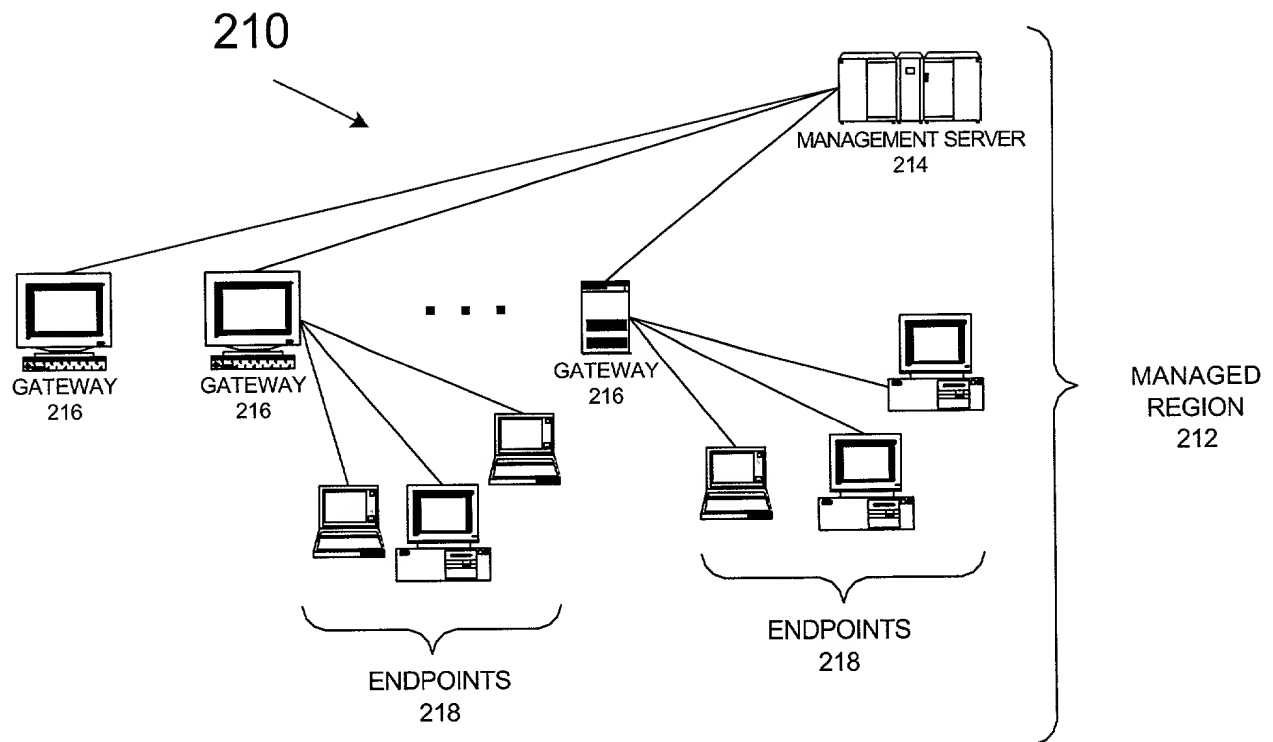


Figure 2A

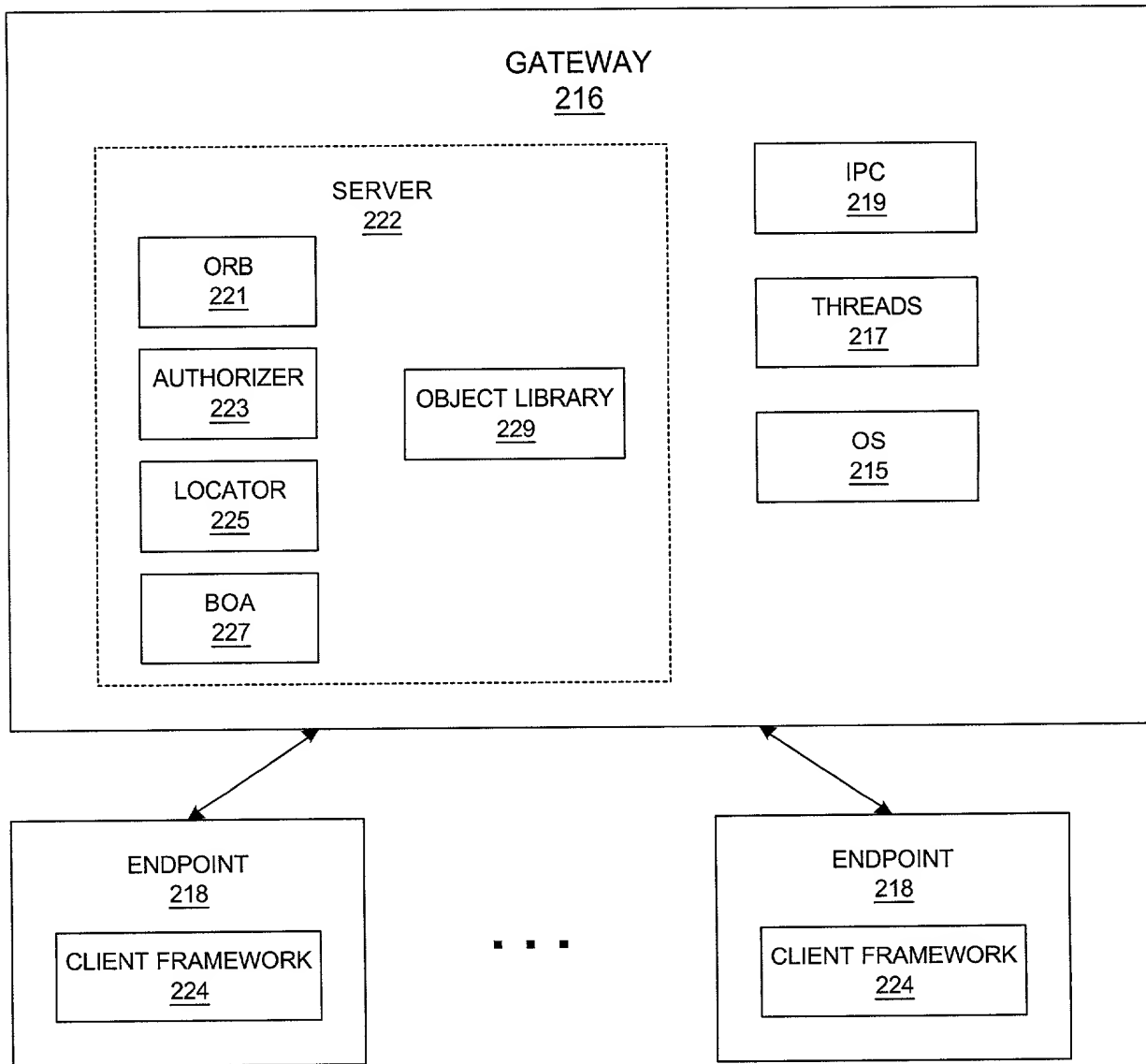


Figure 2B

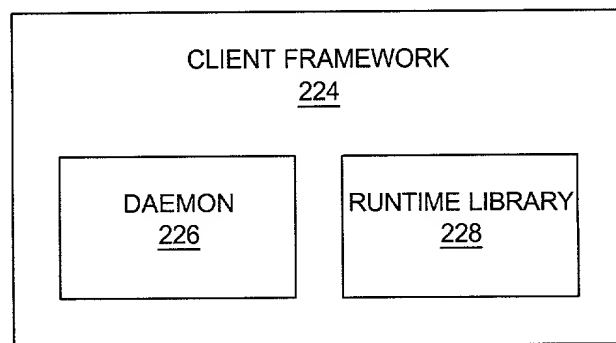


Figure 2C

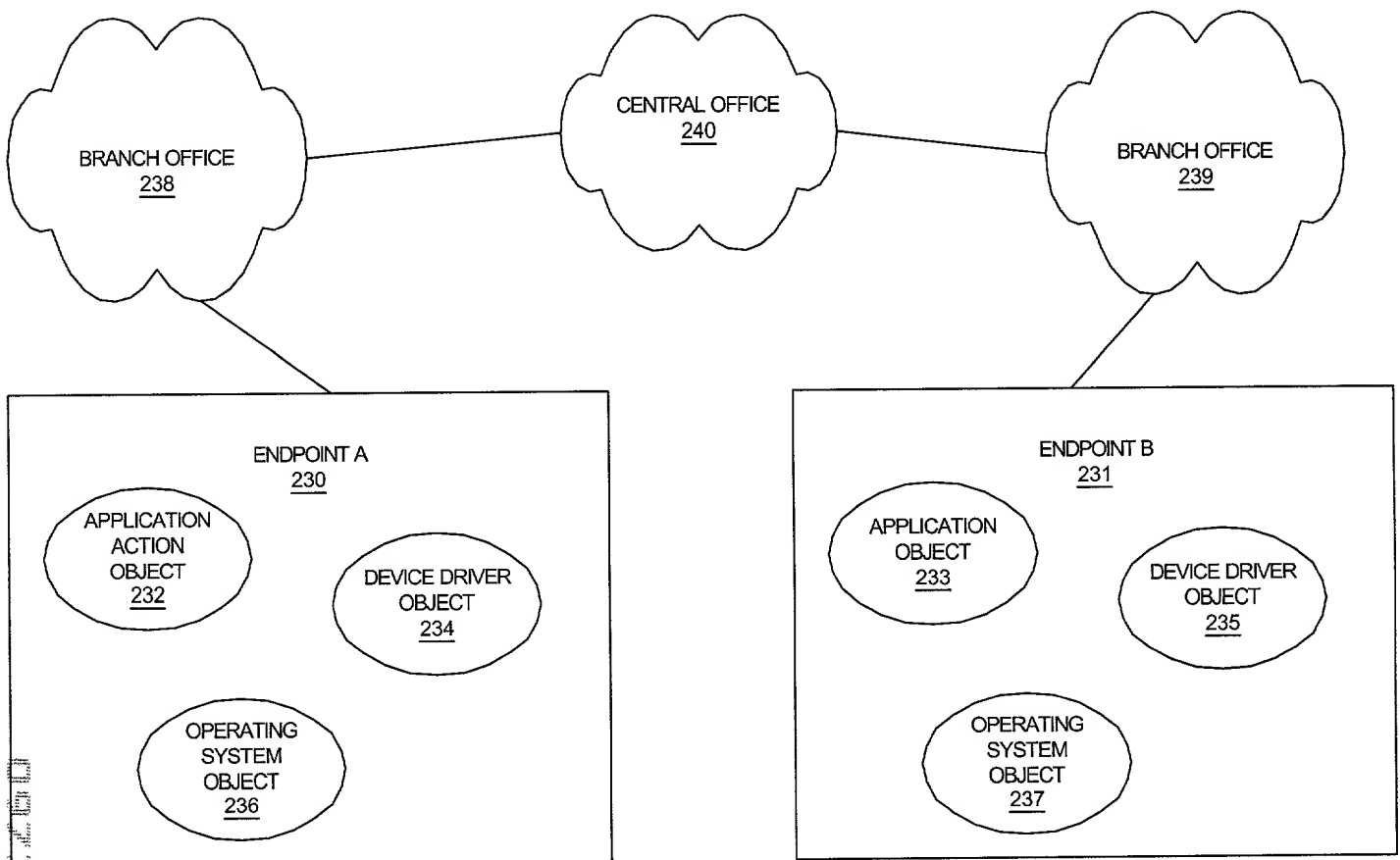


Figure 2D

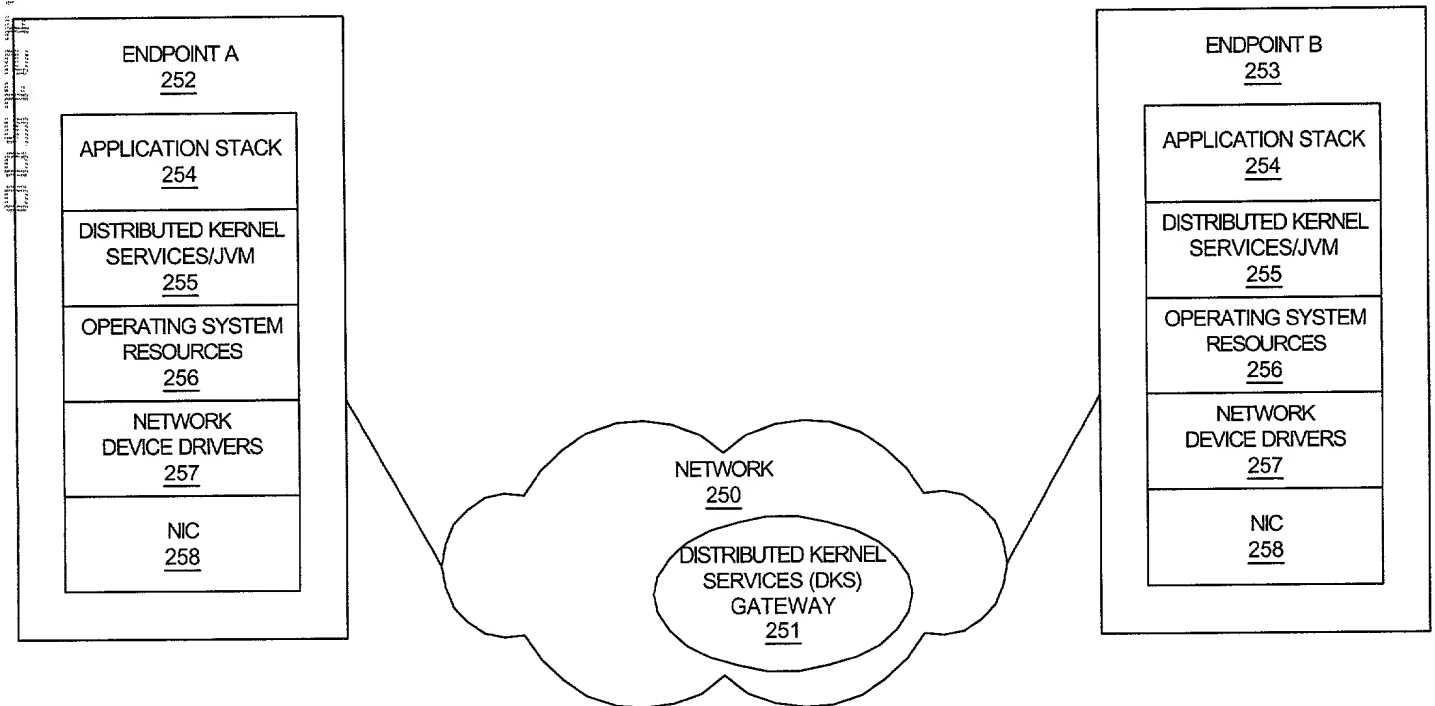


Figure 2E

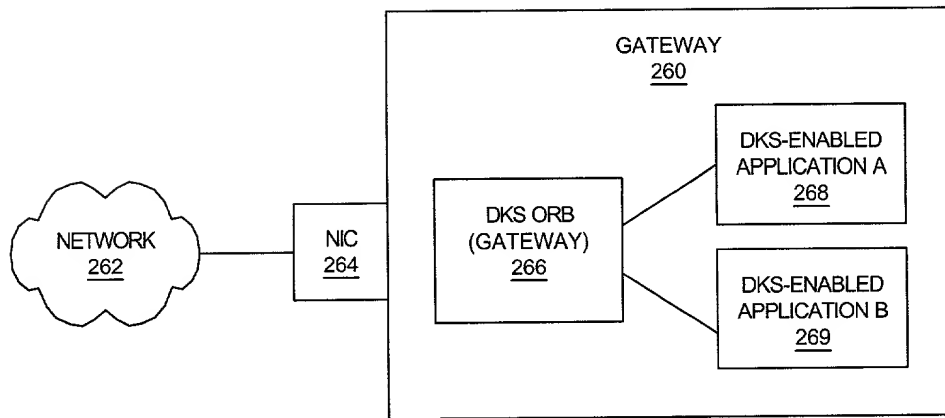


Figure 2F

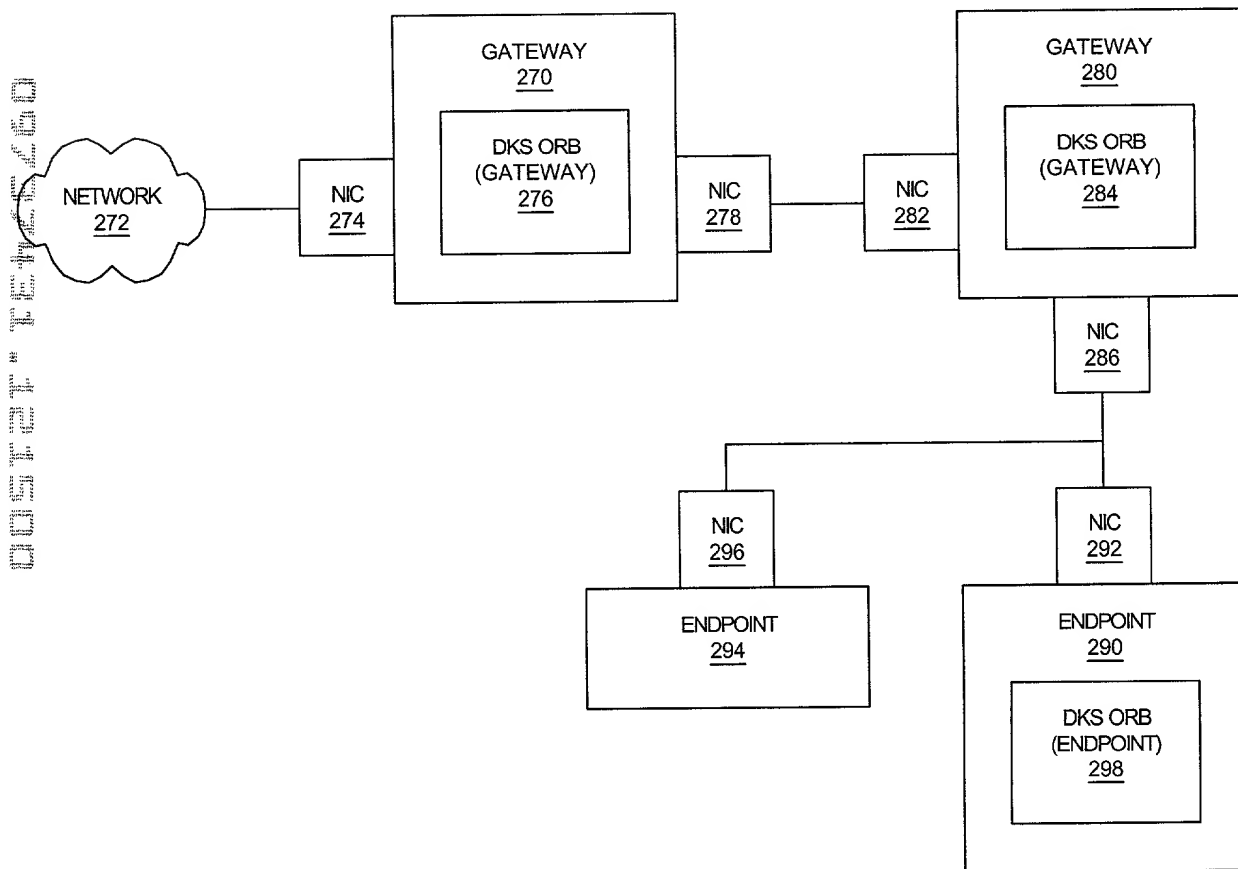


Figure 2G

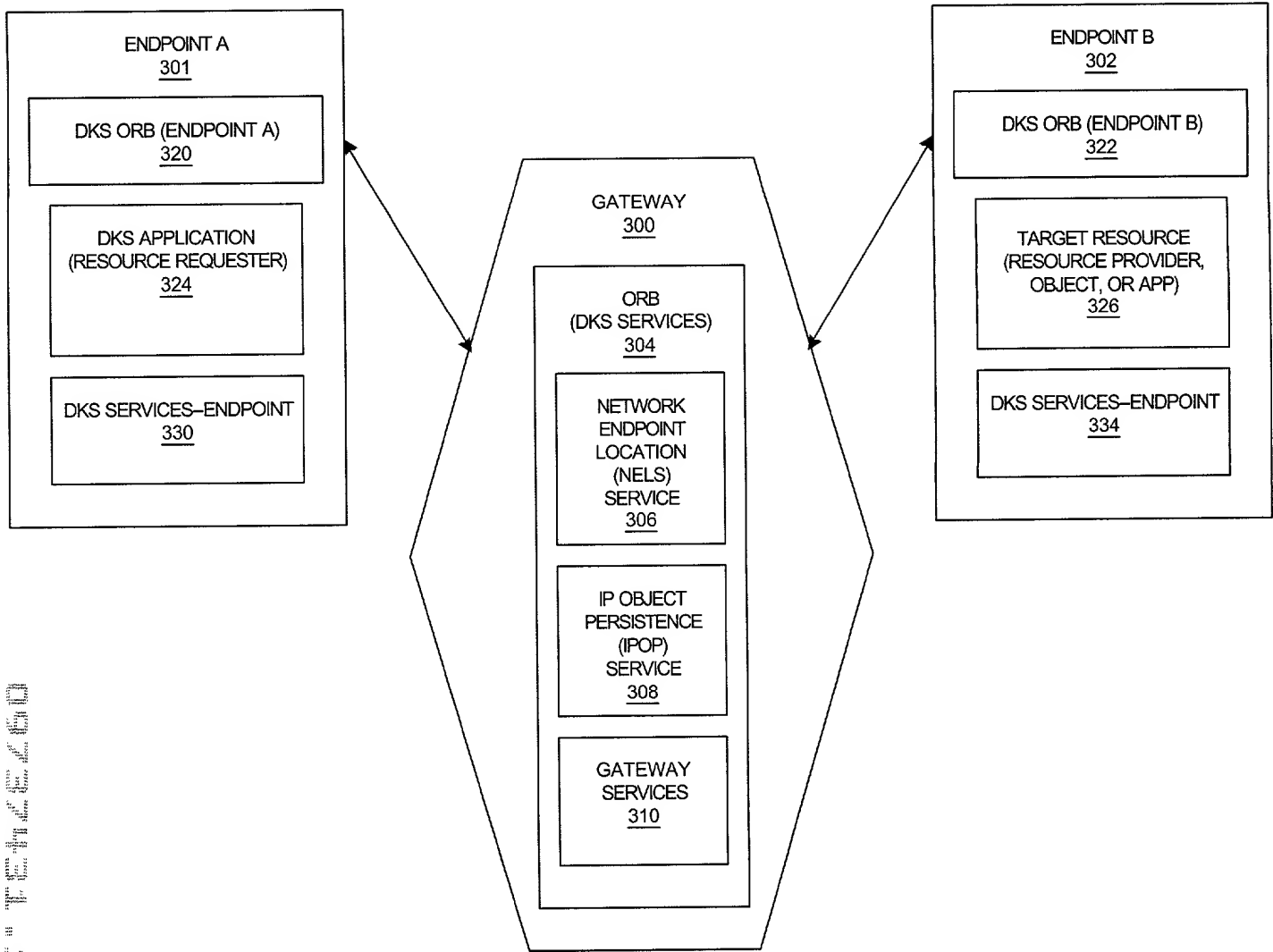


Figure 3

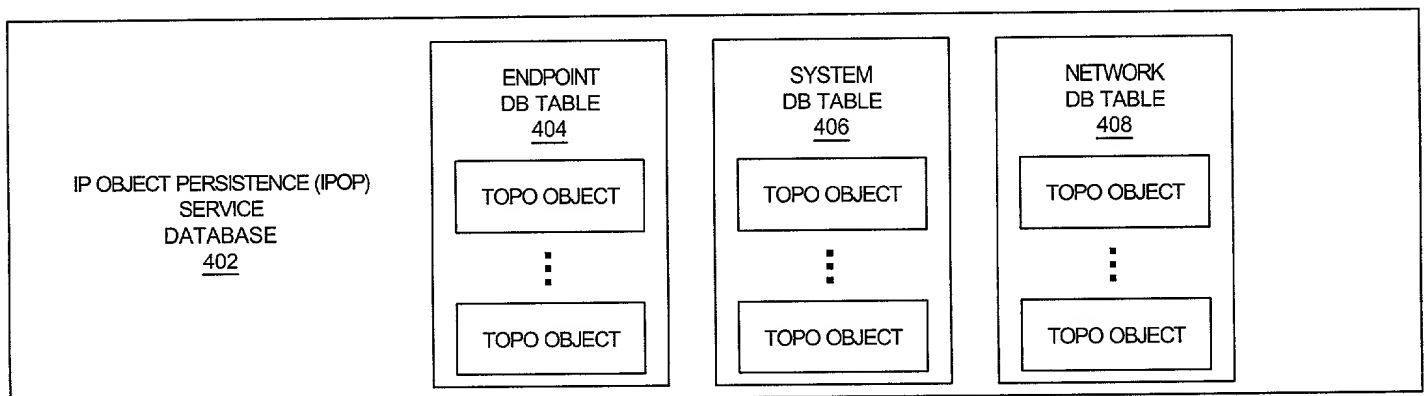


Figure 4

500

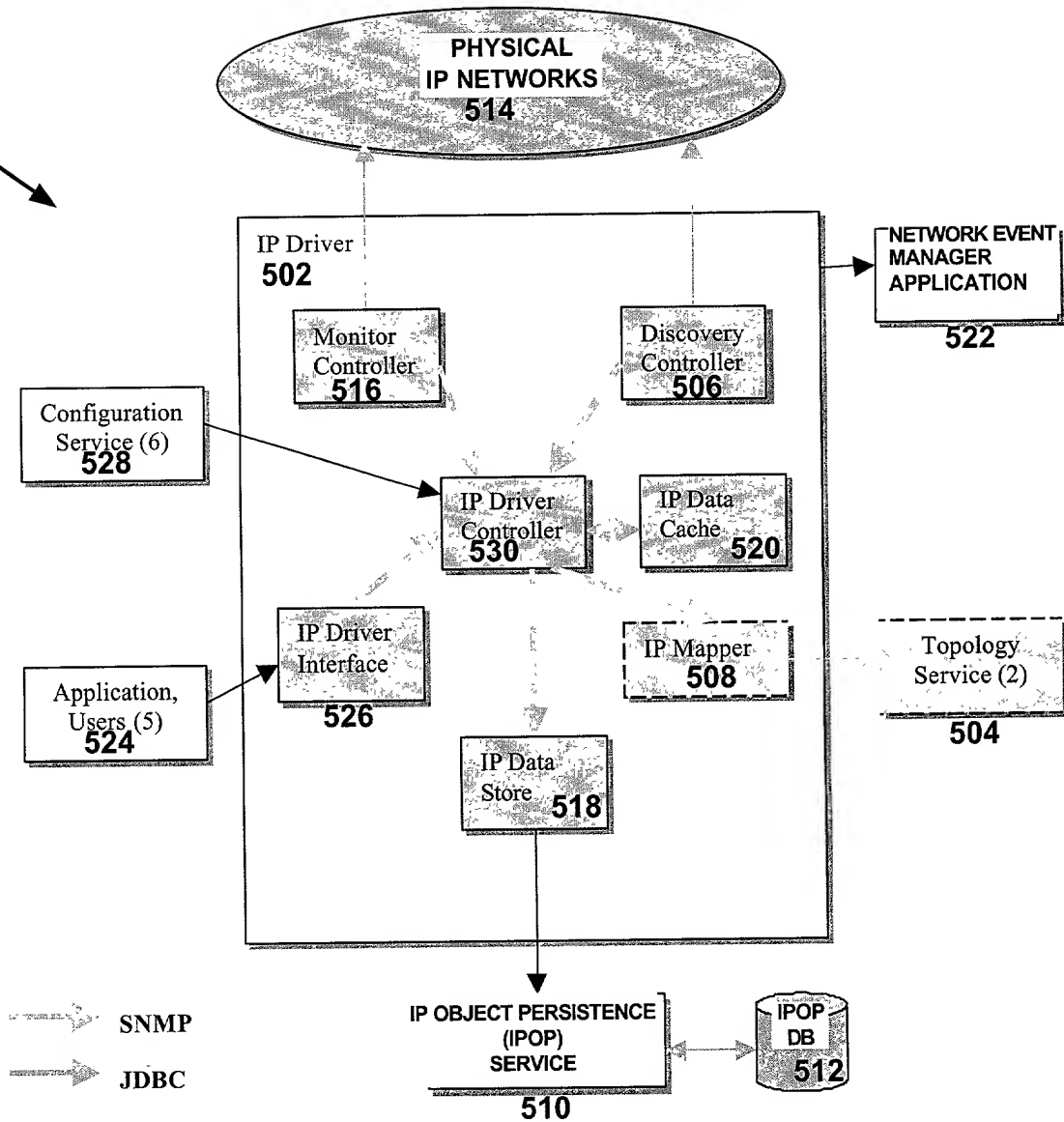


Figure 5A

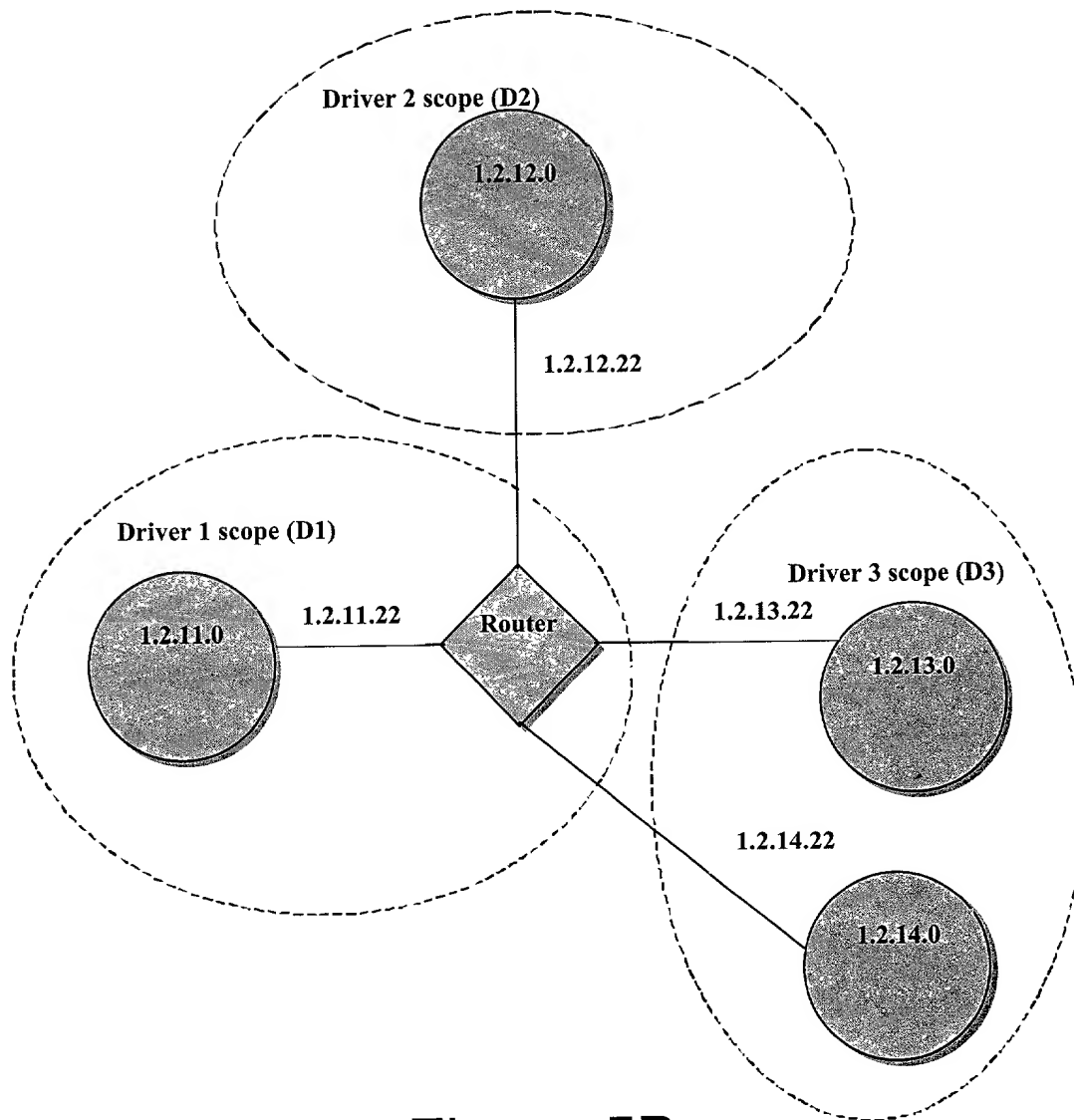


Figure 5B

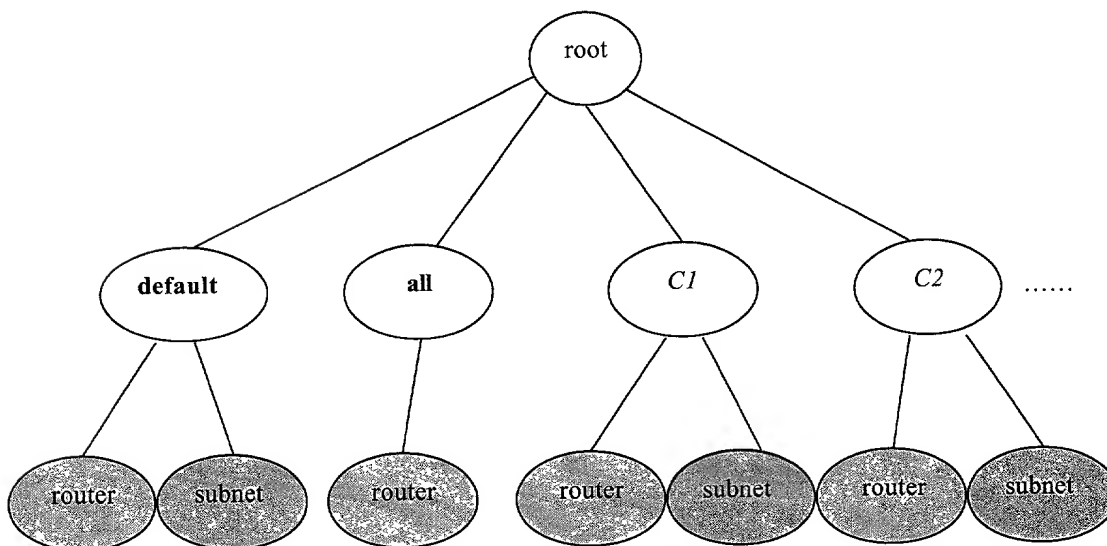


Figure 5C

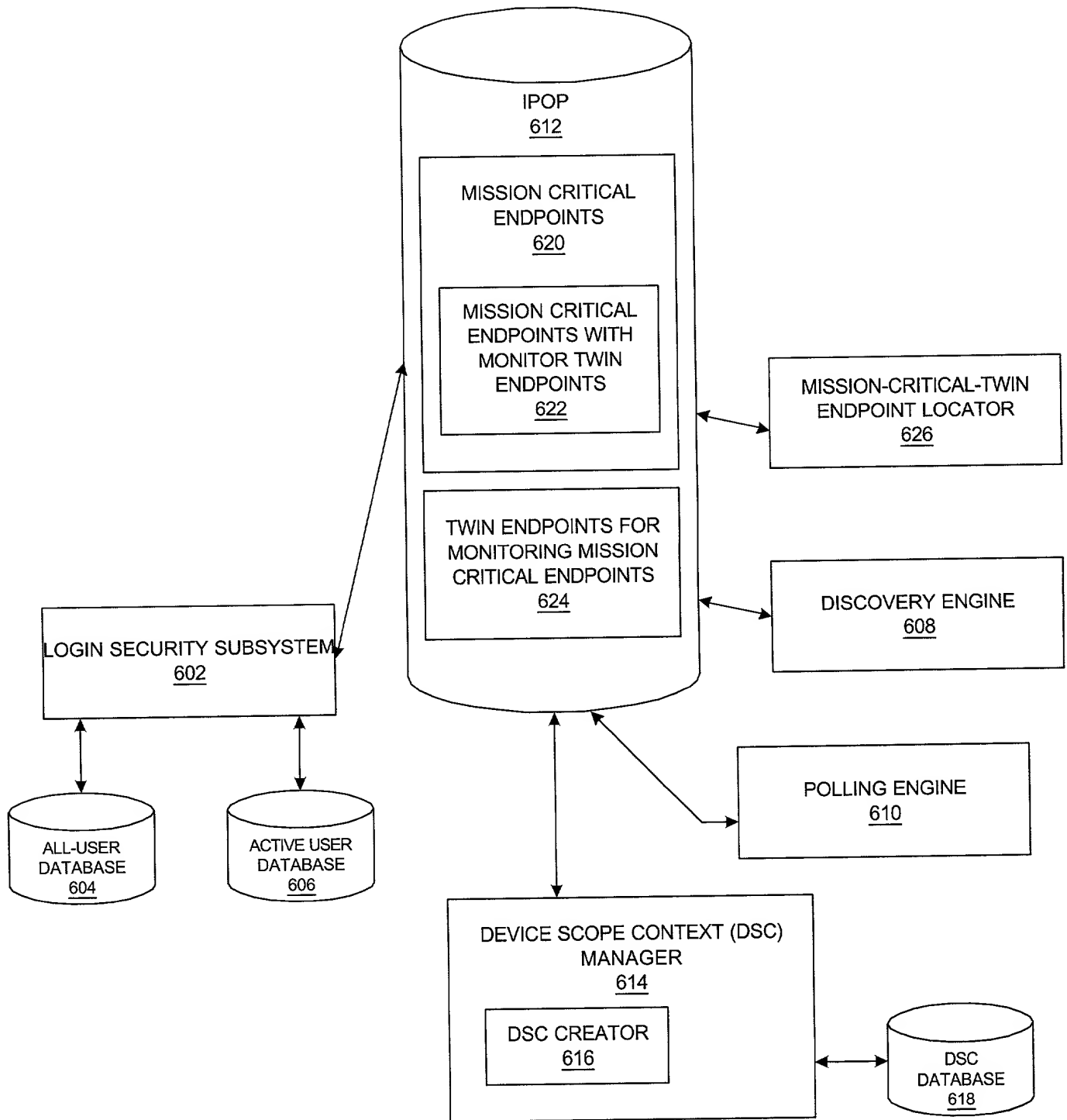


Figure 6

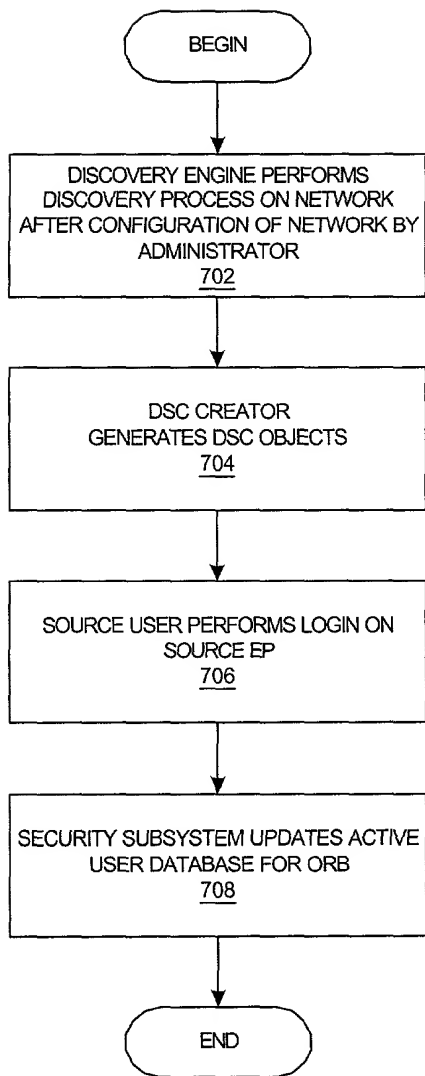


Figure 7A

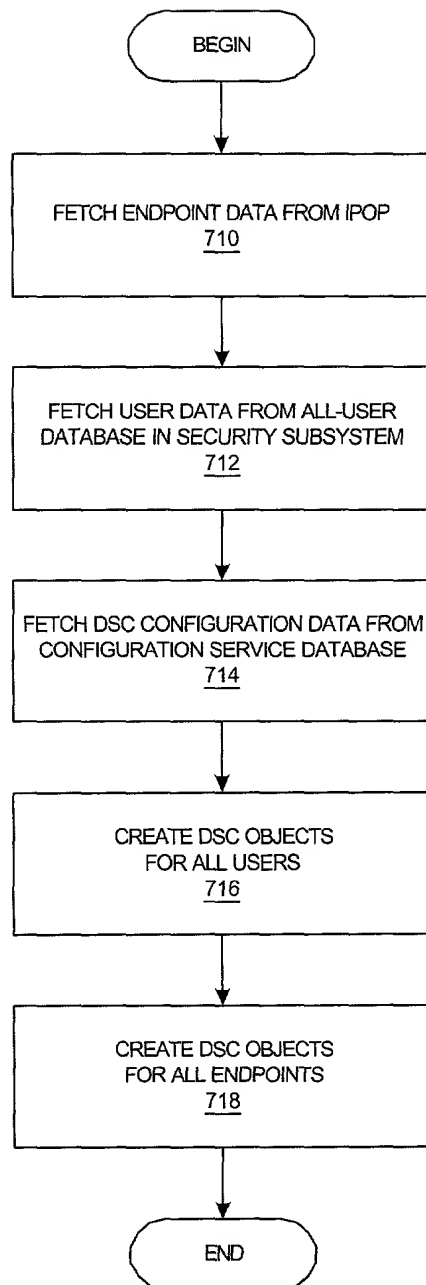


Figure 7B

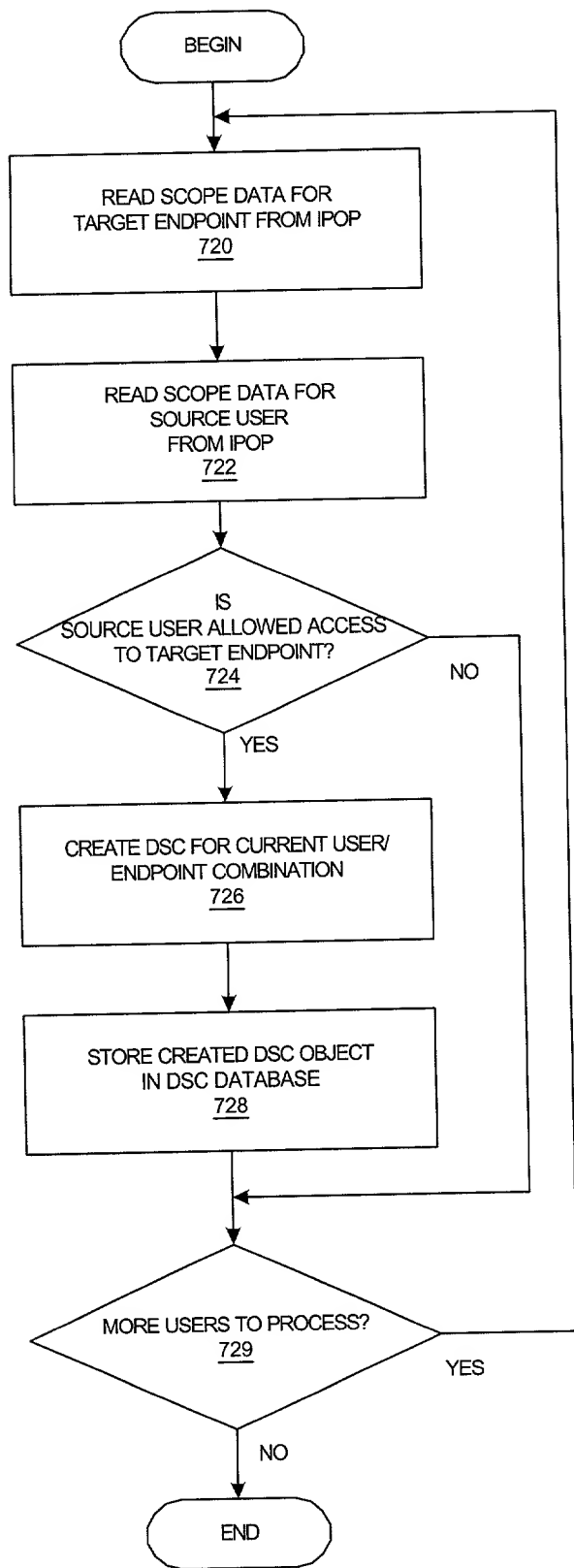


Figure 7C

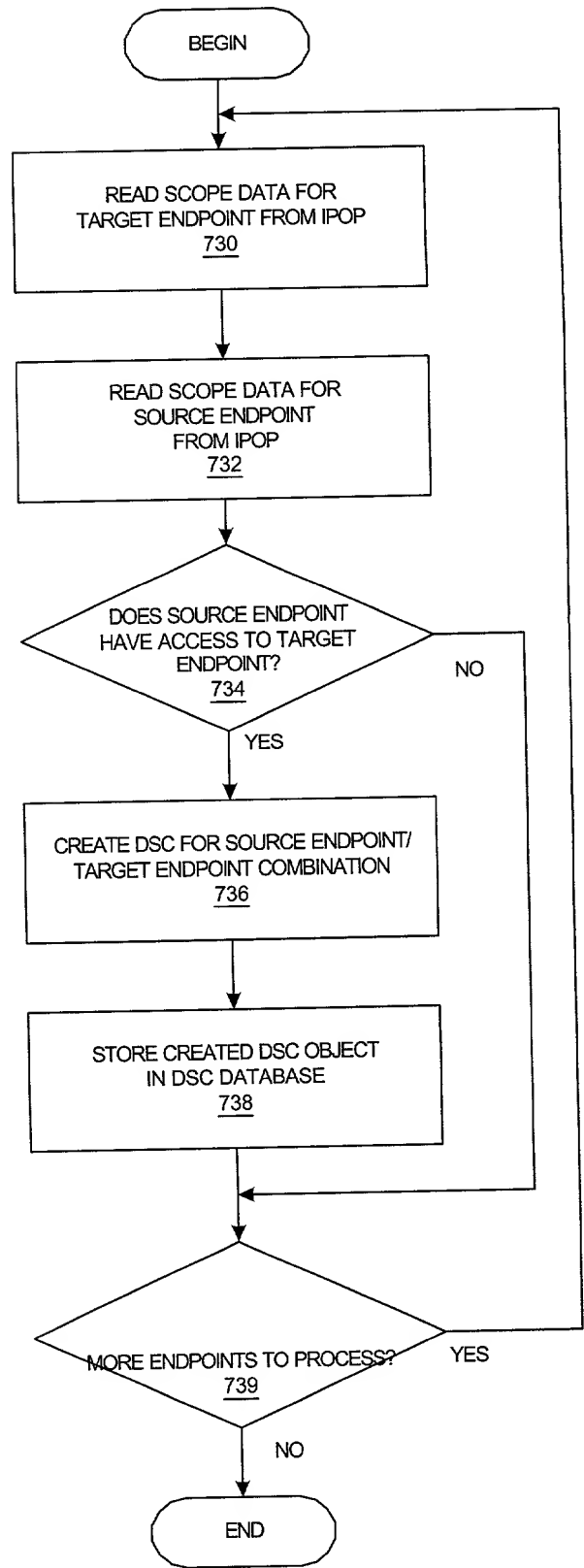


Figure 7D

Figure 1 is a screenshot of a "Network Management Application" window. The window has a title bar with standard OS controls (minimize, maximize, close) in the top right corner. The main content area is divided into two sections. The top section, labeled "ADAPTIVE MONITORING SETTINGS", contains a "POLLING INTERVAL" of "15" minutes (indicated by reference numeral 804). Below this are two radio button options: "SOURCE USER" (selected, indicated by 805) and "SOURCE ENDPOINT" (indicated by 807). The "SOURCE USER" option is associated with a text field containing "JOHN.ADAMS" and a dropdown arrow (indicated by 806). The "SOURCE ENDPOINT" option is associated with a text field containing "CUSTOMER1.IDAHO.BOISE.DEV3498" and a dropdown arrow (indicated by 808). The bottom section, labeled "PRIMARY DSC", contains two radio button options: "BY USER" (selected, indicated by 812) and "BY ENDPOINT" (indicated by 814). This section is enclosed in a box labeled 810. To the right of the "PRIMARY DSC" section are two buttons: "SET" (indicated by 816) and "CLEAR" (indicated by 818). A reference numeral 802 points to the area between the two main sections. A large reference numeral 800 with an arrow points to the left side of the window frame.

Figure 8A

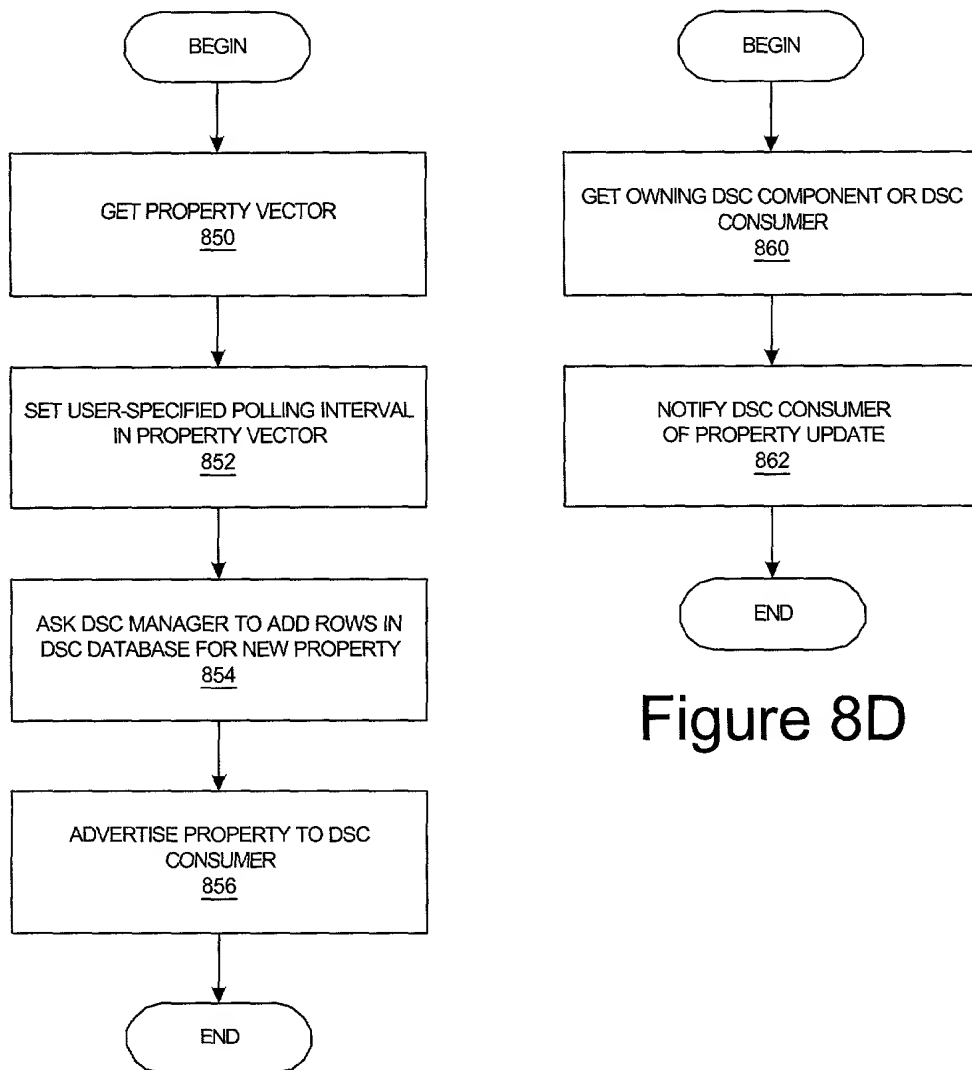


Figure 8D

Figure 8C

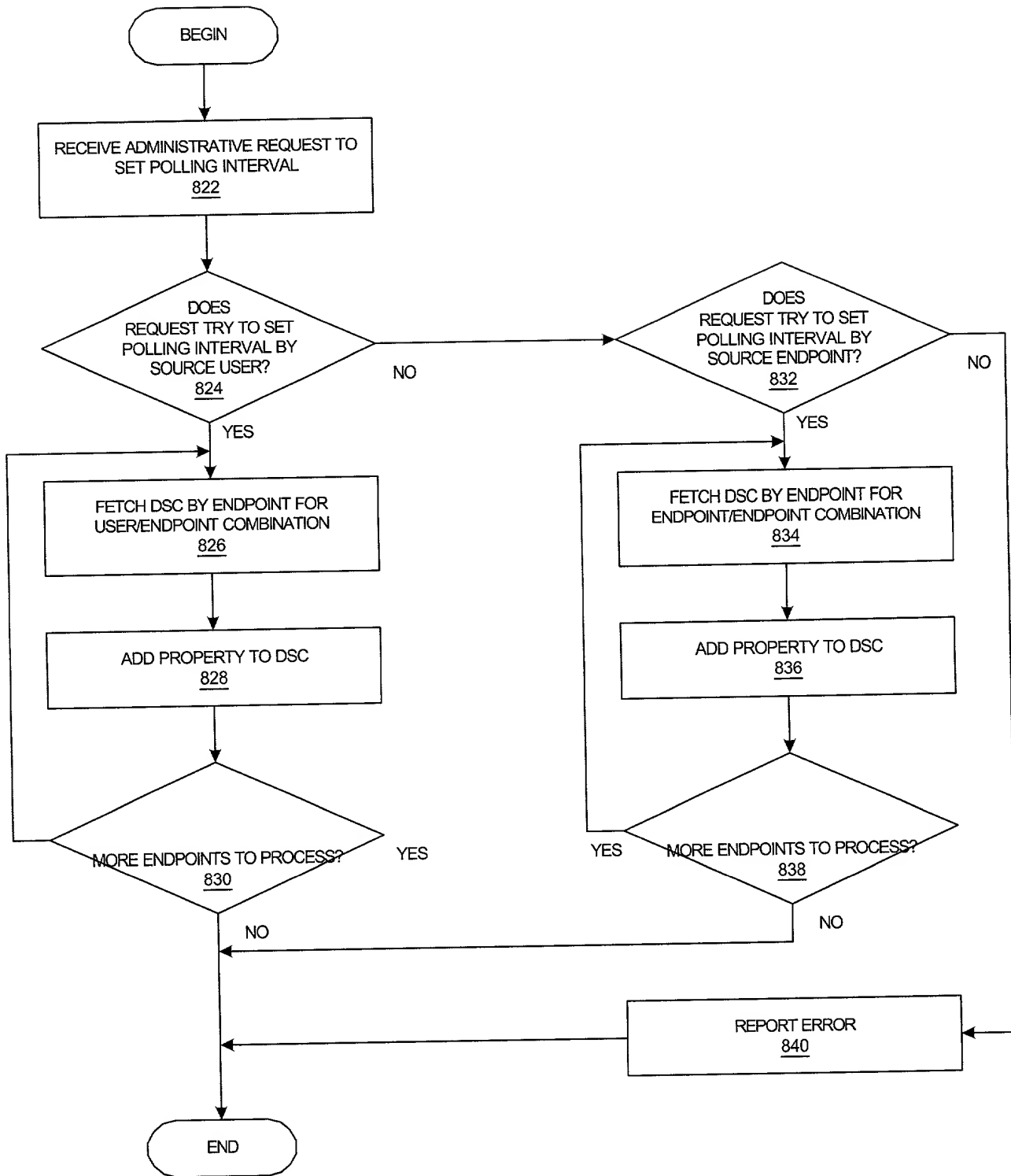
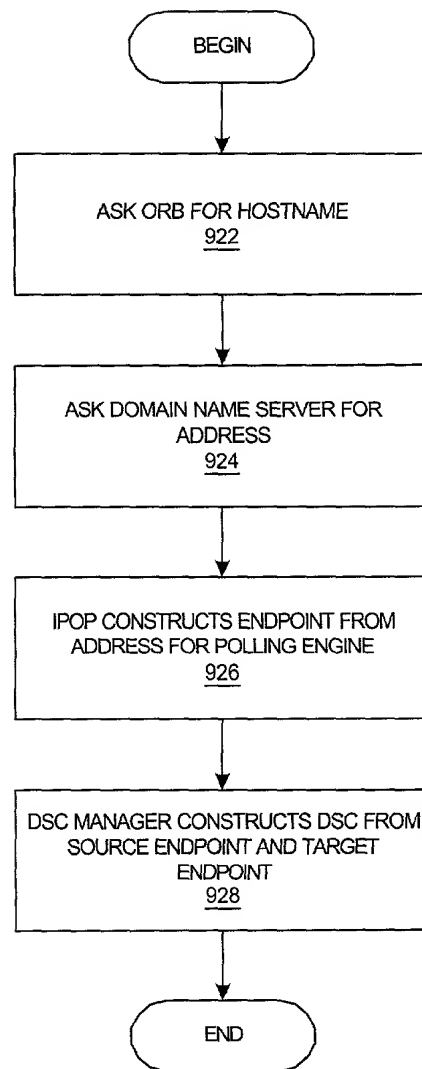


Figure 8B



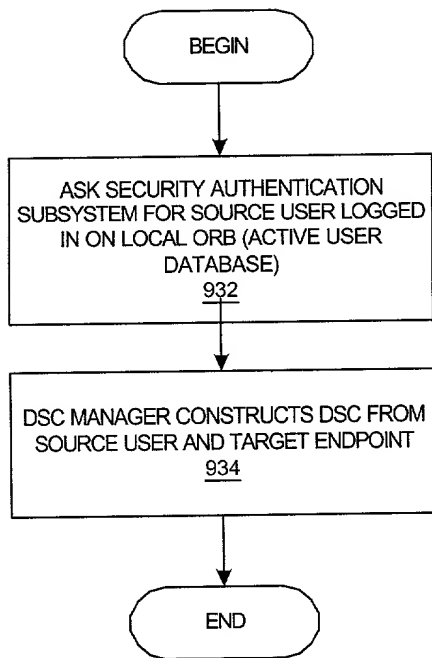


Figure 9C

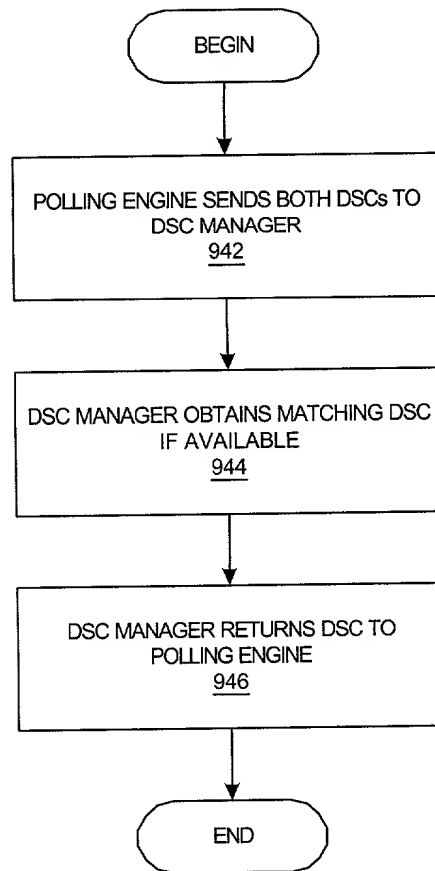


Figure 9D

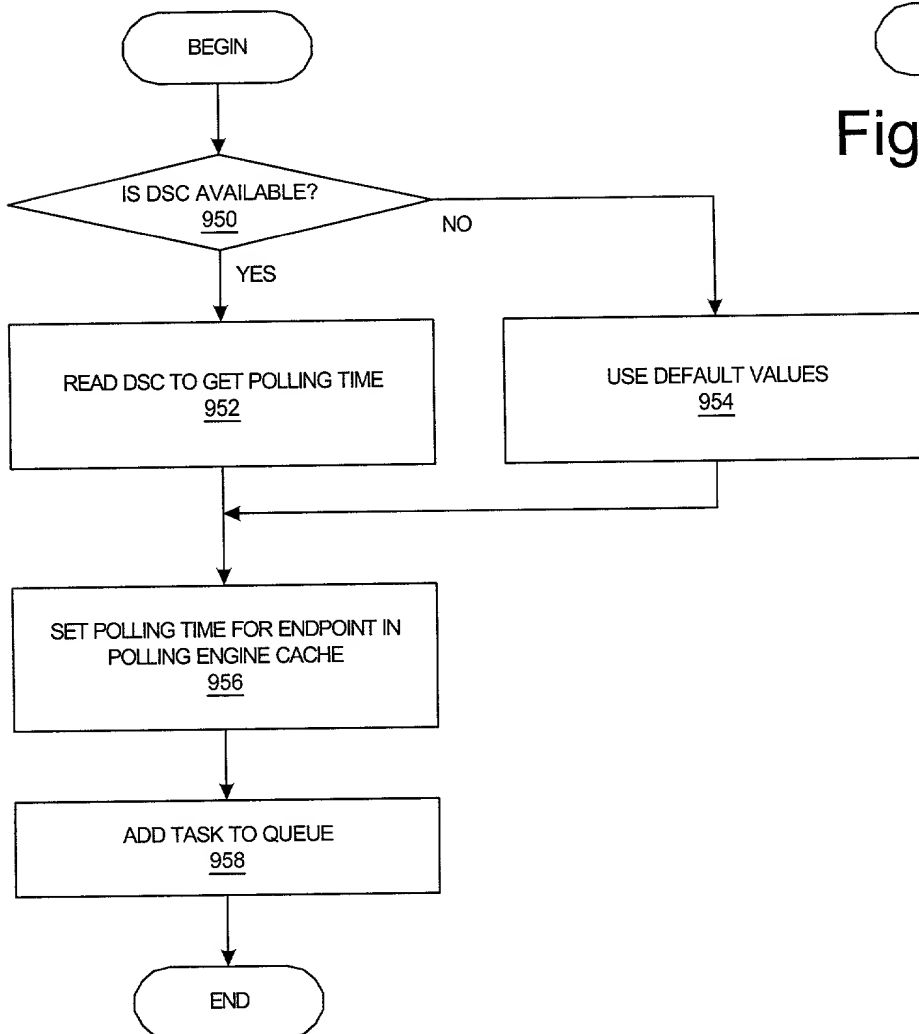


Figure 9E

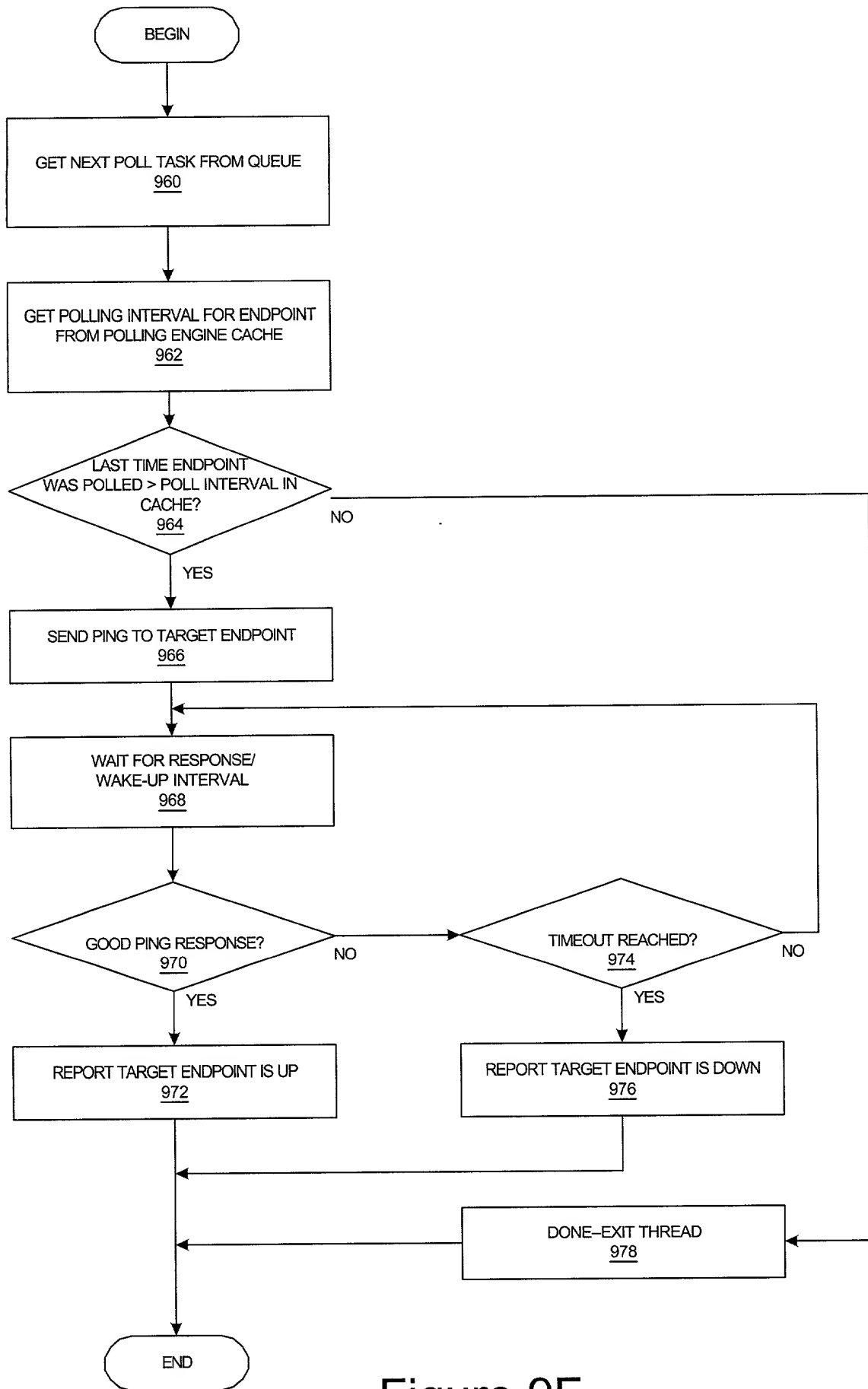


Figure 9F

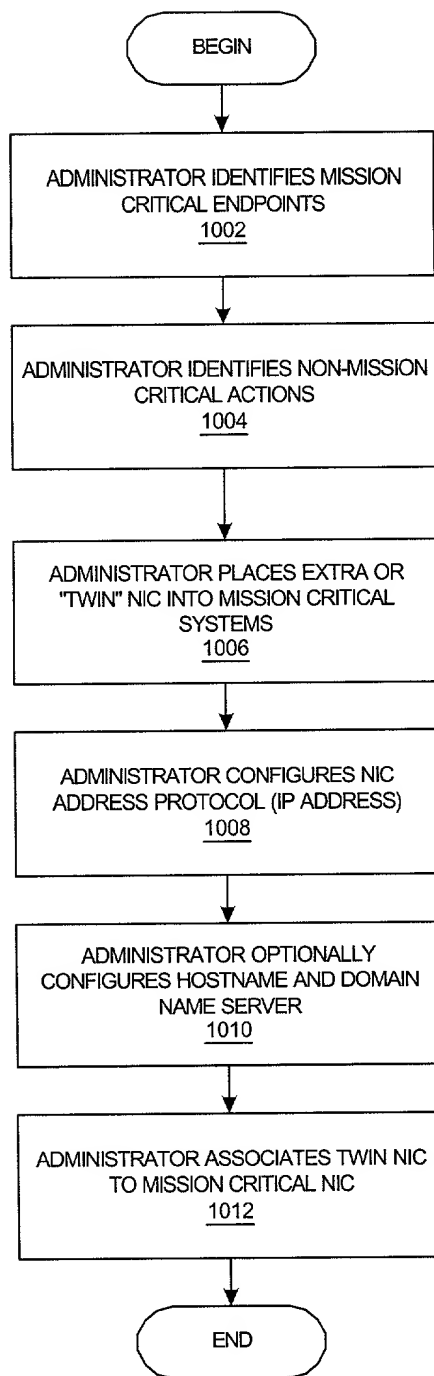


Figure 10A

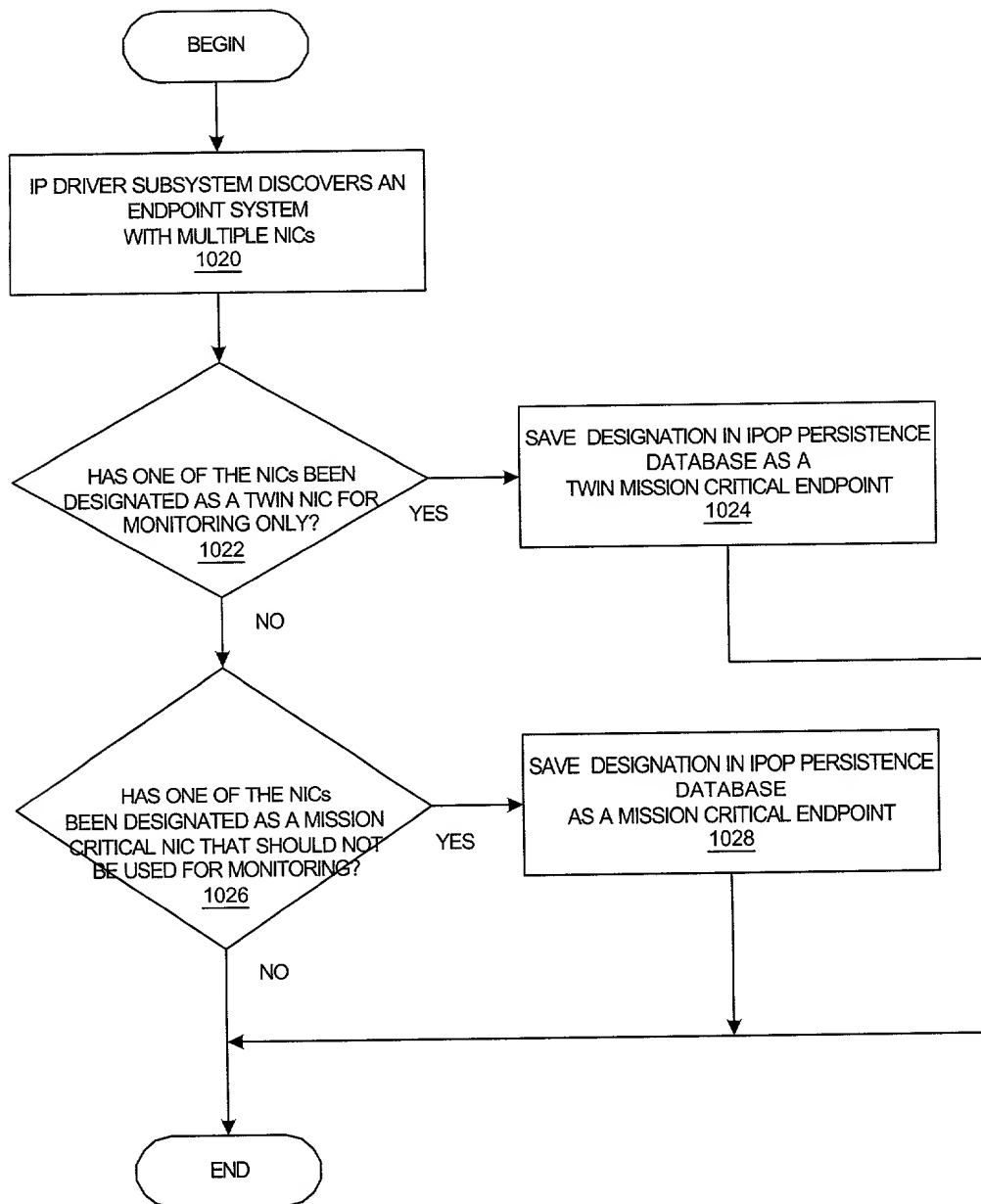


Figure 10B

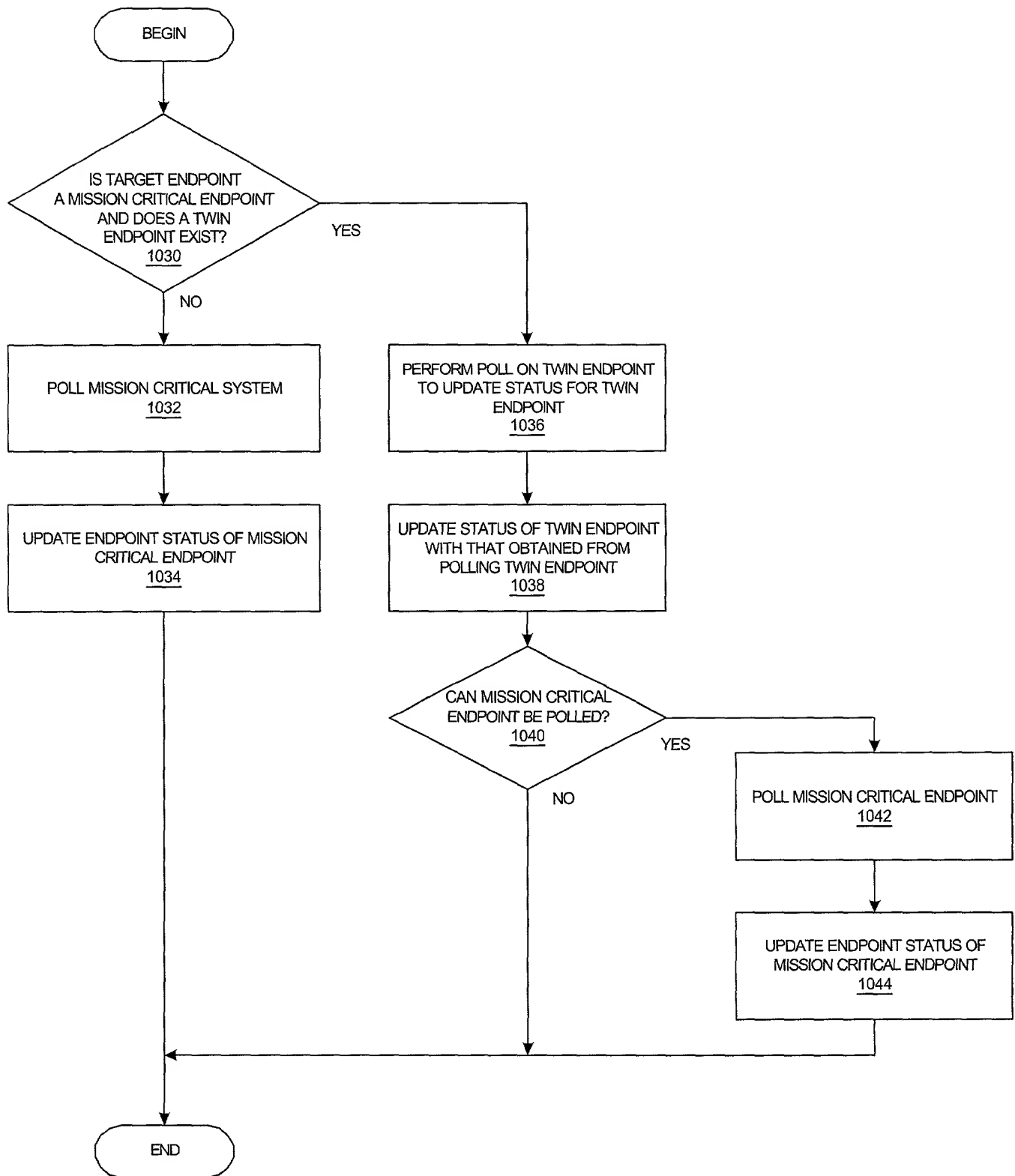


Figure 10C

20/23
AUS9-2000-0704-US1

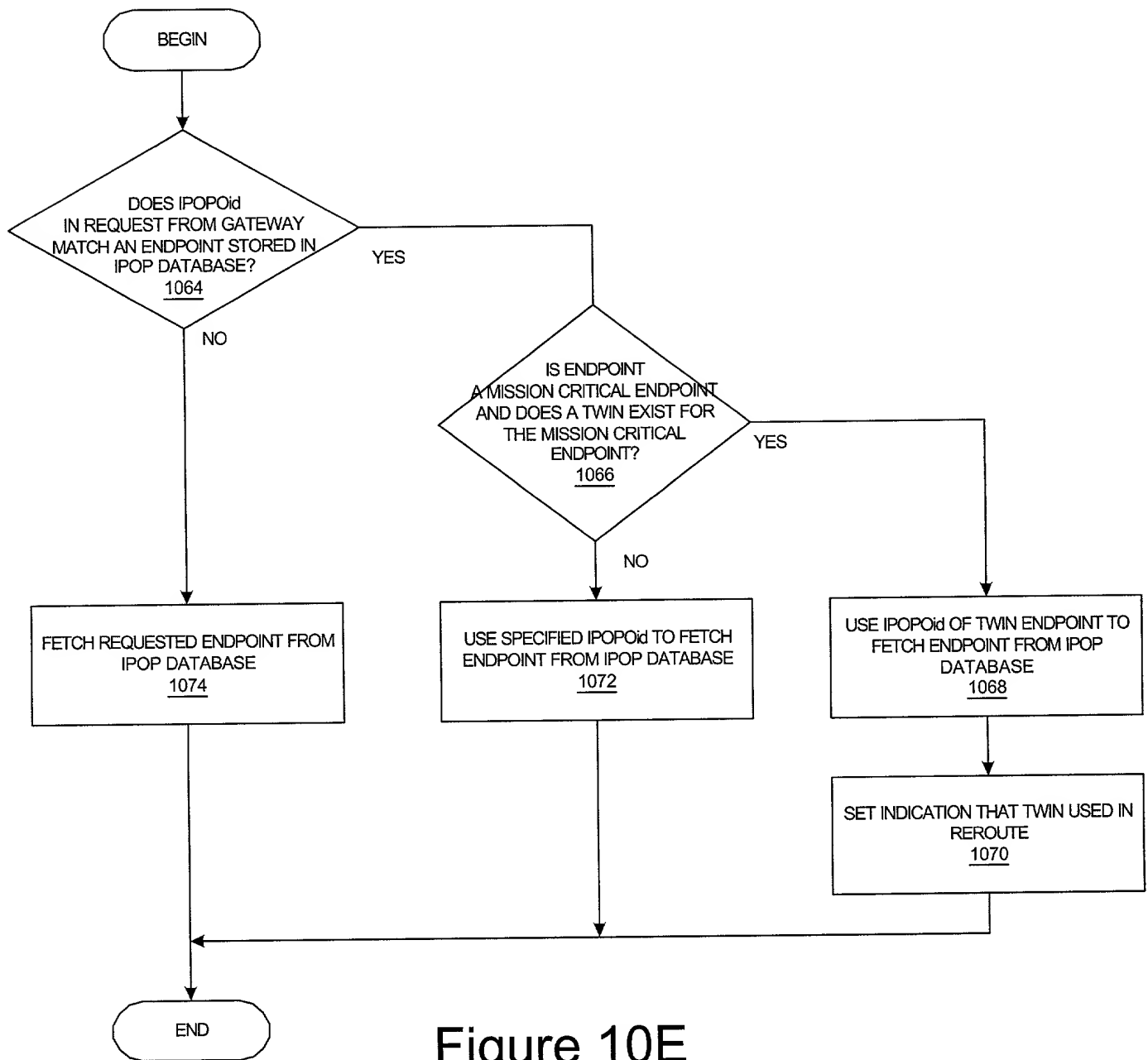


Figure 10E

1090

Network Management Application

MISSION CRITICAL TWIN ASSIGNMENT--MISSION CRITICAL ENDPOINT: 7.17.13.11 1091

ENDPOINT TO USE AS TWIN:

☐ MAC ADDRESS: 1093 1096 1097

☒ 1092 VPN NUMBER: 1094 IP ADDRESS: 1095

The screenshot shows a window titled 'Network Management Application'. Inside, there's a section for 'MISSION CRITICAL TWIN ASSIGNMENT' with a specific endpoint '7.17.13.11'. Below this, there's a form to select an endpoint to use as a twin. Two options are available: 'MAC ADDRESS' (unchecked) and 'VPN NUMBER' (checked). Each option has a corresponding input field. To the right of the input fields are 'SET' and 'CLEAR' buttons.

Figure 10F

```

CLASS ACTION_OBJECT {

    // CONSTRUCTOR
    ACTION_OBJECT( LONG IPADDRESS, SHORT VIRTUALPRIVATENETWORKADDRESS )
        THROWS BADADDRESS ...

    .
    .
    .
    VOID PERFORMACTION( ) // EXECUTES ACTION METHOD

    .
    .
    .
}

```

Figure 11A

```

CLASS APPLICATION_ACTION_OBJECT EXTENDS ACTION_OBJECT {

    boolean IsMissionCriticalAction; // TRUE = USED TO PERFORM ENTERPRISE-RELATED WORK
                                     // FALSE = USED TO PERFORM MONITORING OPERATIONS
                                     //      OR OTHER NON-REVENUE PRODUCING ACTION

    boolean TwinUsedinReroute        // TRUE = IPOP HAS REPLACED REQUESTED ADDRESS WITH
                                     //      AN ADDRESS THAT CAN BE USED FOR
                                     //      NON-MISSION CRITICAL ACTION

    .
    .
    .
}

```

Figure 11B

```

Public Class Endpoint {

    //public variables
    long          EObjectID;    //ID to object (both private and public network addresses)
    InetAddress    EIPAddress;  //physical network address (private or public)
    long          EPVPN;        //virtual private network ID

    //get/set of variables
    public long    getObjectID( ) { ... }
    public InetAddress getIPAddress( ) { ... }
    public long    getVPN( ) { ... }

}

```

Figure 11C

```

Class TwinMissionCriticalEndpoint extends Endpoint {
.
.
.
    IPOPOid    missionCriticalEndpoint; // Mission critical endpoint that is used to gather status
    IPOPOid    missionCriticalSystem;  // Mission critical system

    long        endpointStatus;
    long        twinEndpointStatus;

.
.
.
}

```

Figure 11D

```

Class MissionCriticalEndpoint extends Endpoint {
.
.
.
    boolean    useForMonitoring;    // TRUE = endpoint can be used for monitoring, polling,
                                   //      and other system-management-type resources

.
.
.
}

```

Figure 11E